What is claimed is:

[Claim 1] 1. In a database system, a method for providing automated encryption support for column data, the method comprising:

defining Structured Query Language (SQL) extensions for creating and managing column encryption keys, and for creating and managing database tables with encrypted column data;
receiving an SQL statement specifying creation of a particular column encryption key;
receiving an SQL statement specifying creation of a database table having particular column data encrypted with said particular column encryption key; and
in response to a subsequent database operation that requires the particular column data that has been encrypted, automatically decrypting the particular column data for use by the database operation.

[Claim 2] 2. The method of claim 1, wherein columns that are not specified to be encrypted are stored in unencrypted format, for minimizing encryption overhead.

[Claim 3] 3. The method of claim 1, wherein the automated encryption support operates as an internal built-in feature of the database system, without use of an add-on library.

[Claim 4] 4. The method of claim 1, wherein the SQL statement specifying creation of a particular encryption key is received from a user serving as a system security officer.

[Claim 5] 5. The method of claim 4, wherein the SQL statement specifying creation of a database table may be received from a user other than the system security officer.

[Claim 6] 6. The method of claim 1, wherein the SQL statement specifying creation of a particular encryption key comprises a CREATE ENCRYPTION KEY command.

[Claim 7] 7. The method of claim 6, wherein the CREATE ENCRYPTION KEY command includes:

```
CREATE ENCRYPTION KEY keyname
    [AS DEFAULT] [FOR algorithm]
    [WITH [KEYLENGTH keysize]
        [PASSWD passphrase]
        [INIT_VECTOR [RANDOM | NULL]]
        [PAD [RANDOM | NULL]]]
```

as its syntax.

[Claim 8] 8. The method of claim 1, wherein the SQL statement specifying creation of a database table having particular column data encrypted comprises a CREATE TABLE command that allows specification of one or more columns to be encrypted.

[Claim 9] 9. The method of claim 8, wherein the CREATE TABLE command includes:

```
CREATE TABLE tablename
  (colname1 datatype [encrypt [with [db.[owner].]keyname],
  colname2 datatype [encrypt [with [db.[owner].]keyname])
```

as its syntax.

[Claim 10] 10. The method of claim 1, further comprising:

receiving an SQL statement specifying alteration of a previously-created database table so as to encrypt particular column data.

[Claim 11] 11. The method of claim 10, wherein the SQL statement specifying alteration of a previously created database table comprises an ALTER TABLE command.

[Claim 12] 12. The method of claim 11, wherein the ALTER TABLE command includes:

```
ALTER TABLE tablename MODIFY column_name
  [[datatype] [null|not null]]
  [decrypt | encrypt [with [db.[owner].]keyname]]
```

as its syntax.

[Claim 13] 13. The method of claim 1, wherein the encryption support works transparently with existing database applications.

[Claim 14] 14. The method of claim 1, wherein the database system includes a database server and one or more database clients, and wherein method steps implementing the encryption support are embodied at the database server.

[Claim 15] 15. The method of claim 1, wherein the database system includes a back-end server tier and a middleware tier, and wherein method steps implementing the encryption support are embodied at the back-end server tier.

[Claim 16] 16. The method of claim 1, further comprising:

after creation of the particular column encryption key, protecting the particular column encryption key with a user-supplied password.

[Claim 17] 17. The method of claim 16, wherein the user-supplied password must be supplied before the system allows use of the particular column encryption key for database operations.

[Claim 18] 18. The method of claim 17, wherein the user-supplied password is supplied using a SET ENCRYPTION PASSWD command.

[Claim 19] 19. The method of claim 18, wherein the SET ENCRYPTION PASSWD command includes:

SET ENCRYPTION PASSWD password FOR keyname
as its syntax.

[Claim 20] 20. The method of claim 17, wherein a user seeking to decrypt column data must supply said user-supplied password and must have necessary database privileges before decrypting the column data with the particular column encryption key.

[Claim 21] 21. The method of claim 20, wherein the user-supplied password is supplied using a SET ENCRYPTION PASSWD command.

[Claim 22] 22. The method of claim 1, further comprising:

providing a command to grant decryption permission to others.

[Claim 23] 23. The method of claim 22, wherein the command to grant decryption permission includes:

GRANT DECRYPT ON table.column TO user_or_role_list
as its syntax.

[Claim 24] 24. The method of claim 1, wherein the database system internally stores in encrypted format any column encryption keys that have been created.

[Claim 25] 25. The method of claim 1, wherein the database system stores encrypted column data internally as variable binary (VARBINARY) data.

[Claim 26] 26. The method of claim 1, wherein the database system presents users a user-defined field type for column data that has been encrypted, even though the column data is stored internally as variable binary data.

[Claim 27] 27. The method of claim 1, wherein the database system preserves any user-defined data type for the particular column data so that the database system employs a correct data type for processing queries and returning query results.

[Claim 28] 28. The method of claim 27, wherein the database system stores the user-defined data type for the particular column data in a system catalog of the database system.

[Claim 29] 29. The method of claim 1, wherein the particular column encryption key created comprises a symmetric encryption key.

[Claim 30] 30. The method of claim 1, wherein a single column encryption key is used for each column to be encrypted.

[Claim 31] 31. The method of claim 1, wherein a single column encryption key may be shared by multiple columns to be encrypted.

[Claim 32] 32. The method of claim 1, wherein the particular column encryption key is itself encrypted to a key-encrypting key constructed from a user-supplied password.

[Claim 33] 33. The method of claim 32, wherein the particular column encryption key is itself stored on disk in encrypted format using Advanced Encryption Standard (AES) encryption.

[Claim 34] 34. The method of claim 32, wherein the user-supplied password may comprise a hex literal.

[Claim 35] 35. The method of claim 32, wherein the user-supplied password is itself transformed into a symmetric encryption key, using a random salt, internal static data, and SHA-1 hashing algorithm.

[Claim 36] 36. The method of claim 1, wherein said Structured Query Language (SQL) extensions for creating and managing column encryption keys include a clause for instructing the database system to create a default key for encrypting columns.

[Claim 37] 37. A database system providing automated encryption support for column data, the system comprising:

a parser that supports Structured Query Language (SQL) extensions for creating and managing column encryption keys, and for creating and managing database tables with encrypted column data; and
an execution unit, operating in response to SQL statements parsed by the parser, for creating a particular column encryption key, for creating a database table having particular column data encrypted with said particular column encryption key, and for automatically decrypting the particular column data for

use by a subsequent database operation that requires the particular column data that has been encrypted.

[Claim 38] 38. The system of claim 37, wherein columns that are not specified to be encrypted are stored in unencrypted format, for minimizing encryption overhead.

[Claim 39] 39. The system of claim 37, wherein the automated encryption support operates as an internal built-in feature of the database system, without use of an add-on library.

[Claim 40] 40. The system of claim 37, wherein the SQL statement specifying creation of a particular encryption key is received from a user serving as a system security officer.

[Claim 41] 41. The system of claim 40, wherein the SQL statement specifying creation of a database table may be received from a user other than the system security officer.

[Claim 42] 42. The system of claim 37, wherein the SQL statement specifying creation of a particular encryption key comprises a CREATE ENCRYPTION KEY command.

[Claim 43] 43. The system of claim 42, wherein the CREATE ENCRYPTION KEY command includes:

```
CREATE ENCRYPTION KEY keyname
  [AS DEFAULT] [FOR algorithm]
  [WITH [KEYLENGTH keysize]
    [PASSWD passphrase]
    [INIT_VECTOR [RANDOM | NULL]]
```

```
      [PAD [RANDOM | NULL]]]
```
as its syntax.

[Claim 44] 44. The system of claim 37, wherein the SQL statement specifying creation of a database table having particular column data encrypted comprises a CREATE TABLE command that allows specification of one or more columns to be encrypted.

[Claim 45] 45. The system of claim 44, wherein the CREATE TABLE command includes:

```
CREATE TABLE tablename
  (colname1 datatype [encrypt [with [db.[owner].]keyname],
  colname2 datatype [encrypt [with [db.[owner].]keyname])
```
as its syntax.

[Claim 46] 46. The system of claim 37, further comprising:

a module for receiving an SQL statement specifying alteration of a previously created database table so as to encrypt particular column data.

[Claim 47] 47. The system of claim 46, wherein the SQL statement specifying alteration of a previously created database table comprises an ALTER TABLE command.

[Claim 48] 48. The system of claim 47, wherein the ALTER TABLE command includes:

```
ALTER TABLE tablename MODIFY column_name
  [[datatype] [null|not null]]
  [decrypt | encrypt [with [db.[owner].]keyname]]
```
as its syntax.

[Claim 49] 49. The system of claim 37, wherein the encryption support works transparently with existing database applications.

[Claim 50] 50. The system of claim 37, wherein the database system includes a database server and one or more database clients, and wherein the encryption support is provided by the database server.

[Claim 51] 51. The system of claim 37, wherein the database system includes a back-end server tier and a middleware tier, and wherein the encryption support is provided by the back-end server tier.

[Claim 52] 52. The system of claim 37, wherein the system protects the particular column encryption key with a user-supplied password.

[Claim 53] 53. The system of claim 52, wherein the user-supplied password must be supplied before the system allows use of the particular column encryption key for database operations.

[Claim 54] 54. The system of claim 53, wherein the user-supplied password is supplied using a SET ENCRYPTION PASSWD command.

[Claim 55] 55. The system of claim 54, wherein the SET ENCRYPTION PASSWD command includes:

SET ENCRYPTION PASSWD password FOR keyname
as its syntax.

[Claim 56] 56. The system of claim 53, wherein a user seeking to decrypt column data must supply said user-supplied password and must have necessary database privileges before decrypting the column data with the particular column encryption key.

[Claim 57] 57. The system of claim 37, further comprising:

providing a command to grant decryption permission to others.

[Claim 58] 58. The system of claim 57, wherein the command to grant decryption permission includes:

GRANT DECRYPT ON table.column TO user_or_role_list
as its syntax.

[Claim 59] 59. The system of claim 37, wherein the database system internally stores in encrypted format any column encryption keys that have been created.

[Claim 60] 60. The system of claim 37, wherein the database system stores encrypted column data internally as variable binary (VARBINARY) data.

[Claim 61] 61. The system of claim 37, wherein the database system presents users a user-defined field type for column data that has been encrypted, even though the column data is stored internally as variable binary data.

[Claim 62] 62. The system of claim 37, wherein the database system preserves any user-defined data type for the particular column data so that the database system employs a correct data type for processing queries and returning query results.

[Claim 63] 63. The system of claim 62, wherein the database system stores the user-defined data type for the particular column data in a system catalog of the database system.

[Claim 64] 64. The system of claim 37, wherein the particular column encryption key created comprises a symmetric encryption key.

[Claim 65] 65. The system of claim 37, wherein a single column encryption key is used for each column to be encrypted.

[Claim 66] 66. The system of claim 37, wherein the particular column encryption key is itself encrypted to a key-encrypting key constructed from a user-supplied password.

[Claim 67] 67. The system of claim 66, wherein the particular column encryption key is itself stored on disk in encrypted format using Advanced Encryption Standard (AES) encryption.

[Claim 68] 68. The system of claim 66, wherein the user-supplied password may comprise a hex literal.

[Claim 69] 69. The system of claim 66, wherein the user-supplied password is itself transformed into a symmetric encryption key, using a random salt, static internal data and SHA-1 hashing algorithm.

[Claim 70] 70. The system of claim 37, wherein said Structured Query Language (SQL) extensions for creating and managing column encryption keys include a clause for instructing the database system to create a default key for encrypting columns.

[Claim 71] 71. In a database system, a method for encrypting column data, the method comprising:

in response to a first query language statement, creating an encryption key for encrypting a particular column of a database table;
in response to a second query language statement, encrypting the particular column using said encryption key; and
during a subsequent database operation requiring column data from the particular column, automatically decrypting the column data for use by the database operation.

[Claim 72] 72. The method of claim 71, further comprising:

assigning privileges to users for creating an encryption key for encrypting column data.

[Claim 73] 73. The method of claim 72, further comprising:

in response to a request to create an encryption key from a particular user, determining whether the particular user has sufficient privileges to create an encryption key.

[Claim 74] 74. The method of claim 71, wherein the encryption key is itself encrypted to a key-encrypting key constructed from a user-supplied password.

[Claim 75] 75. The method of claim 74, wherein the encryption key is encrypted using Advanced Encryption Standard (AES) encryption.

[Claim 76] 76. The method of claim 74, wherein the user-supplied password may comprise a hex literal.

[Claim 77] 77. The method of claim 74, wherein the user-supplied password is itself transformed into a symmetric encryption key, using a random salt, static internal data and SHA-1 hashing algorithm.

[Claim 78] 78. The method of claim 71, wherein the database system stores encrypted column data internally as variable binary (VARBINARY) data.

[Claim 79] 79. The method of claim 71, wherein columns of the database table that are not specified to be encrypted are stored in unencrypted format.

[Claim 80] 80. The method of claim 71, wherein the system implements said first and second statements as SQL extensions for creating and managing encryption keys and for creating and managing database tables with encrypted column data.

[Claim 81] 81. The method of claim 80, wherein said SQL extensions include a CREATE ENCRYPTION KEY command for creating an encryption key.

[Claim 82] 82. The method of claim 81, wherein said CREATE ENCRYPTION KEY command includes attributes specifying an encryption key name and a user-supplied password.

[Claim 83] 83. The method of claim 80, wherein said SQL extensions include a CREATE TABLE command having an attribute that allows specification of at least one column to be encrypted.

[Claim 84] 84. The method of claim 83, wherein said CREATE TABLE command syntax includes attributes specifying a table name, one or more columns to be encrypted, and an encryption key name.

[Claim 85] 85. The method of claim 71, wherein said second query language statement includes a request specifying alteration of a previously-created table so as to encrypt particular column data.

[Claim 86] 86. The method of claim 71, wherein a user subsequently requiring use of the encrypted column data must provide a user-supplied password for unlocking the encryption key for the particular column.

[Claim 87] 87. The method of claim 71, further comprising:

receiving an SQL statement specifying creation of a default key encryption password.

[Claim 88] 88. The method of claim 87, wherein the SQL statement specifying creation of a default key encryption password specifies a default password value that is encrypted by a system stored procedure, for storage in a system table of a particular database.

[Claim 89] 89. The method of claim 71, further comprising:

receiving an SQL statement specifying creation of an encryption keypair.

[Claim 90] 90. The method of claim 89, wherein the SQL statement specifying creation of an encryption keypair comprises a CREATE ENCRYPTION KEYPAIR command.

[Claim 91] 91. The method of claim 90, wherein the CREATE ENCRYPTION KEYPAIR command includes:

```
CREATE ENCRYPTION KEYPAIR keypairname
  [FOR algorithm]
  [WITH [KEYLENGTH keysize]
  [PASSWD passphrase | LOGIN_PASSWD]
```
as its syntax.

[Claim 92] 92. The method of claim 71, further comprising:

receiving an SQL statement specifying alteration of a particular encryption key or keypair.

[Claim 93] 93. The method of claim 71, further comprising:

receiving an SQL statement specifying dropping a particular encryption key or keypair.

[Claim 94] 94. The method of claim 71, further comprising:

receiving an SQL statement granting rights to a particular encryption key or keypair.

[Claim 95] 95. The method of claim 94, further comprising:

receiving an SQL statement revoking said rights that have been granted to a particular encryption key or keypair.

[Claim 96] 96. The method of claim 94, wherein the said rights granted for the particular encryption key or keypair comprise SELECT query execution rights, for selecting encrypted data.

[Claim 97] 97. The method of claim 94, wherein the said rights granted for the particular encryption key or keypair comprise ALTER query execution rights, for altering the encryption key or keypair.

[Claim 98] 98. A computer-readable medium having processor-executable instructions for performing the method of claim 71.

[Claim 99] 99. A downloadable set of processor-executable instructions for performing the method of claim 71.